



ITKMontage: A Software Module for Image Stitching

Dženan Zukić¹ · Michael Jackson² · Dennis Dimiduk² · Sean Donegan³ · Michael Groeber⁴ · Matthew McCormick¹

Received: 11 December 2020 / Accepted: 2 February 2021 / Published online: 8 March 2021
© The Minerals, Metals & Materials Society 2021

Abstract

Stitching partially overlapping image tiles into a montage is a common requirement for materials microscopy. We developed *ITKMontage*, a new module for the open-source *Insight Toolkit (ITK)*, capable of robustly and quickly generating extremely large, high-bit-depth montages within the memory constraints of standard workstations. The phase correlation method is at the core of our pairwise tile registration algorithm. Precise alignment of tiles acquired in typical raster patterns is enhanced with sub-pixel fitting and cropping to overlap. Fast Fourier transform (FFT)-based correlation is improved through a variety of padding methods and an added adjustable bias toward an expected translation. To arrange the tiles into the overall montage, we use global least squares minimization with outlier detection and removal. To blend tiles smoothly, each tile's contribution is weighted by the distance from the tile's edge. Results are demonstrated on several material science data sets and 3D images. Results compare favorably to *ImageJ/Fiji* plugin *Image Stitching*. The tool is integrated into the DREAM.3D software suite for multidimensional, multimodal microstructural data.

Keywords Image · Registration · Stitching · Montage

Introduction

A cornerstone of materials science is that the internal structure of materials strongly determines properties and performance. Characterization of such internal structure is critical for quantitative analysis of this structure-properties relationship; however, the structure features of interest often span a hierarchy of length scales that could extend well into the macroscopic range in both 2 and 3 dimensions, complicating the characterization process. This behavior is observed for a variety of materials systems, including crystalline materials that are relevant to both geological samples and structural alloys, composite materials, and biological materials, including volumetric rendering of the human body. Materials exhibit long wavelength heterogeneity and develop both unique structure

morphologies (sometimes referred to as the material fabric) and preferred crystal orientations (texture) that can extend to meter-long scales. To characterize these structures, biologists, geologists, and materials scientists make extensive use of various imaging techniques, such as transmission and scanning electron microscopy modes, bright-field optical microscopy, and polarized-light optical microscopy. A common approach for allowing these techniques to characterize regions that are sufficiently large enough to capture relevant hierarchical structure is to collect multiple image units, or tiles, with overlapping local fields of view. These tiles may then be stitched together to form a single montage image that exhibits a field of view far larger than the instrument would be able to capture at the same spatial resolution of the individual tiles. This

✉ Dženan Zukić
dzenan.zukic@kitware.com

¹ Kitware Inc., 101 East Weaver Street, Suite G4, Carrboro, NC 27510, USA

² BlueQuartz Software LLC, Springboro, OH 45066, USA

³ Air Force Research Laboratory, Dayton, OH 45324, USA

⁴ The Ohio State University, Columbus, OH 43210, USA

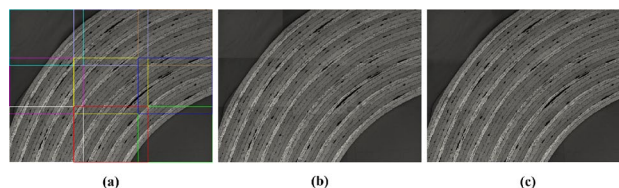


Fig. 1 Data set: an organic matrix composite (data set name: OMC14), 3 × 3 tiles, 12% overlap (a). Montage without smooth blending (b). Final montage, size 3547 × 2666 pixels (c)

montage approach to image collection is frequently used in materials science, and several instrument vendors provide built-in capabilities for collecting such data sets. A small example is given in Fig. 1. If significant field of view is needed to capture the structure scale of interest, or if similarly high local resolution is needed to resolve features of interest, these montage data sets can quickly increase in size, both in terms of the number of collected tiles and the resolution of each tile.

Further, these large data sets may present challenges for analysis and interpretation, particularly if sample preparation or acquisition conditions yield images that are not ideal for automated processing or segmentation. In such cases, human manual segmentation may be needed, and such human intervention is rather untenable for extremely wide area montages, or for the large three-dimensional (3D) data sets that are being increasingly used to characterize structure [7, 20, 25]. Consequently, the analyst benefits from image processing tools that operate on the image montages (or their prepared unit tiles) to bring the image data into a close approximation of ideal conditions. For example, individual image tiles often exhibit a nonlinear background contrast. If images tiles are stitched together without correcting this background contrast, the resulting montage may exhibit characteristic “stitch lines”, as shown in Fig. 1b. The conditions of the montage image can be improved by suitably leveling, smoothing, or blending the contrast, achieving a self-consistent montage field [3, 10, 14].

One of the oldest digital montaging software packages is the ImageJ plugin Image Stitching developed by Preibisch et al. [19]. The Stitching Plugin (2D–5D) is able to reconstruct large images/stacks from an arbitrary number of tiled input images/stacks, making use of the Fourier Shift Theorem that computes all possible translations (x , y , z) between two 2D/3D images at once, yielding the best overlap in terms of the cross correlation measure. If more than two input images/stacks are used the correct placement of all tiles is determined using a global optimization. The stitching is able to align an arbitrary amount of channels, and supports time-lapse registration. To remove brightness differences at the tile borders, nonlinear intensity blending can be applied.

The follow-up package by the same authors, [BigStitcher](#), is still under active development in *beta* status. [BigStitcher](#) allows simple and efficient alignment of multi-tile and multi-angle image data sets, for example acquired by light sheet, widefield or confocal microscopes. The software supports images of almost arbitrary size ranging from very small images up to volumes in the range of many terabytes, that are for example produced when acquiring cleared tissue samples using light sheet microscopy.

[ZetaStitcher](#) [16] is a tool designed to stitch large volumetric images such as those produced by light sheet fluorescence microscopes, on which it is primarily focused.

Its development was started some months before ours, and largely continued concurrently to development of the present work.

Microscopy Image Stitching Tool ([MIST](#)) [4] is being developed at the National Institute of Standards and Technology for rapid and accurate stitching of large 2D time-lapse mosaics. [MIST](#) estimates the mechanical stage model parameters (actuator backlash, and stage repeatability r) from computed pairwise translations and then minimizes stitching errors by optimizing the translations within a $(4r)^2$ square area. [MIST](#) has a performance-oriented implementation utilizing multicore hybrid CPU/GPU computing resources.

The first three of these previous or concurrent packages are licensed under GNU General Public License (either version 2 or 3). This license’s viral copyleft nature makes it unsuitable for incorporation into liberally licensed open-source software, as well as proprietary software. [MIST](#) has a liberal, BSD-like [license](#), but it is 2D-only which somewhat limits its applicability. Furthermore, it is written in Java as a plug-in for ImageJ so its incorporation into other software is not easy.

The [Insight Toolkit \(ITK\)](#) [11, 17] is an open-source, cross-platform library that provides developers with an extensive suite of software tools for image analysis. Developed using open science methodologies, [ITK](#) builds on a proven, spatially oriented architecture for processing, segmentation, and registration of scientific images in two, three, or more dimensions. Its liberal Apache 2.0 license allows incorporation of the code into a wide range of software projects, including proprietary software. We present here a montaging software package implemented as a remote module for [ITK](#) called `ITKMontage`, with a focus on scalability, speed, and accuracy. The montaging tools are also made available in the end-user software suite [DREAM.3D](#) [8], which is specifically targeted toward the needs of the materials community.

Methods

`ITKMontage` takes a list of images in a grid layout and their approximate expected position from the mechanical control system of the microscope. Each pair of adjacent images is registered (Sect. 2.2), and a set of best candidate translations are recorded. The final position of each tile is determined by least squares minimization (Sect. 2.3). The case of non-grid layout is not explicitly handled, but might be accomplished by inserting blank (all-black) tiles to achieve, e.g., circular shape.

Once optimized positions are computed, the tiles can be blended into a single montage image. Overlapping parts of tiles are weighted using distance from the closest

edge to achieve a smooth blend (Fig. 1). The software provides a debugging option during blending that outputs random uniform colors, which is useful for understanding the behavior of the registration or blending process. See Fig. 10(left) for a visualization of this uniform color blending.

Correction of Intensity Inhomogeneity

Pixel intensities for the same material can be uneven across an image as a result of uneven illumination, camera position, or other experimental conditions. It is desirable to correct this intensity inhomogeneity even before registering, but especially before fusing the tiles within a larger montage. We accomplish this by using ITK's `N4BiasFieldCorrectionImageFilter`, that maximizes frequency content within a tile. This bias correction method was contributed by Tustison et al. [23, 24] and is an extension of work by Sled et al. [21].

The algorithm assumes a simple parametric model (Gaussian) for the intensity bias field and does not require segmentation of the underlying image. The basic algorithm iterates between sharpening the intensity histogram of the corrected input image and spatially smoothing those results via a B-spline scalar field estimate of the bias field. A visual example is given in Fig. 2.

Our `example` computes bias field for each tile separately, as we are aiming for generality. Bias field which varies from tile to tile can be due to heterogeneities in the sample that lead to local surface height variation as a result of non-uniform polishing. If a bias field is expected to be the same for every tile, it can be estimated only once in order to save computation time.

Pairwise Registration

We started from an existing ITK software module [1], then updated it to work with modern compilers and added

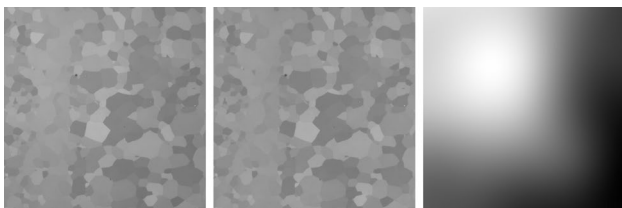


Fig. 2 Example of bias correction on an image of the titanium alloy Ti-7Al. Left: original tile. Middle: corrected tile. Right: estimated multiplicative bias (re-scaled to 0–255 range). A material pixel having intensity 140 in the corrected image might have intensity of 149 near top-left, but only 128 near the bottom right corner in the original image

various code improvements. The updated software module provides a framework for phase correlation registration [12], comprised of the following steps, some of which can be customized:

1. Resampling and padding the image tiles to the same spacing and size. We implemented zero padding, mirror padding, and mirror padding with exponential decay.
2. Compute the fast Fourier transform (FFT) of the two images.
3. Compute the ratio of the two resulting spectra.
4. Apply a Butterworth band-pass filter in the frequency domain. This improves resistance to noise (high-frequency content) and image heterogeneity (low-frequency content), e.g., due to lighting variations in microscopes.
5. Compute the inverse FFT of the cross-power spectrum.
6. Find the maximum peak in the cross-power spectrum and estimate the needed translation. In this step, we optionally restrict peak search to the maximum expected deviation from the initial tile position, suppress the trivial zero solution, and bias the search toward the expected position. We keep a few highest peaks, in case the highest one is judged an outlier by the optimization procedure later on.

Bias toward the expected position is implemented by reducing a correlation factor by multiplication with $e^{-f*(d/s)^2}$, where f is distance penalty factor, d is a pixel's distance from the expected peak location, and s is image size.

Due to finite image resolution and content, the phase correlation peak can be spread across multiple pixels. We therefore allow adjacent peaks to be merged; the default merge distance is 1 pixel.

In real tile image data sets, translation offsets between images are rarely integer multiples of the pixel spacing. To account for this expected discrepancy, we incorporate sub-pixel interpolation as an option during the peak finding. We implemented both parabolic fitting and cosine fitting [18] to determine the sub-pixel peak location. These methods attempt to determine an exact position of an intensity peak in the real cross-correlation surface. Both of these fitting methods assume a function fit to the intensities of the pixel containing the peak, as well as two adjacent pixels (Fig. 3). We also implemented zero-crossing in the frequency domain over the weighted mean phase of the complex phase correlation [22]. After translating the image back according to the integer peak offset, the fractional subpixel offset, δ_x , along a given dimension is estimated from the complex phase correlation in the frequency domain:

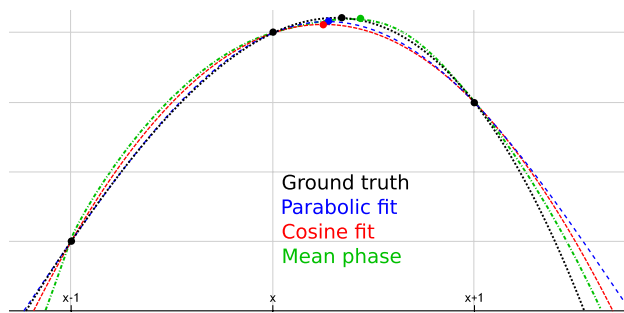


Fig. 3 The three samples ($x - 1, x, x + 1$) around the peak are interpolated with a function (blue, red, green) that does not necessarily match the underlying ground truth (black). The displacement (peaks' projection onto abscissa) from the analytically determined parametric peak will differ from reality depending on how well the chosen parametric function matches the ground truth

$$\delta_x = - \frac{\sum_{k=1}^N \frac{\phi(X_k) |S_k|^2}{\omega_k}}{\sum_{k=1}^N |S_k|^2}$$

where the offset estimated by the phase, $\phi(X_k)$ to angular frequency ω_k is weighted by the power $|S_k|$.

When computing the registration between tile pairs, the input images are cropped to the overlapping region, which we found to significantly increase registration reliability and improve computational performance.

Global Optimization of Tile Positions

Localization of the best tile positions is formulated as a global optimization of an over-determined linear system following [19], formed by one equation for each tile pair i :

$$-c_i p_f + c_i p_m = c_i t_i$$

where c_i is the confidence of the pairwise registration (intensity of the cross-correlation peak), t_i is translation difference to the initial tile position, p_f is position of the 'fixed' (left/top/shallow) tile, p_m is position of the 'moving' (right/bottom/deep) tile. Note: while c_i could cancel out mathematically, the magnitude of coefficients in the linear system affects each equation's residual error in L_2 norm minimization, thus controlling the importance of that equation.

We also add a final equation to constrain the top left tile to have position 0:

$$c_a p_0 = 0$$

where $c_a = \frac{\sum_{i=0}^{n-1} c_i}{n}$ is the average confidence, and n is the number of registration pairs.

The unknown positions p have the same dimension as the translations on the right-hand side, usually 2D or 3D.

As the formulated linear system of equations is very sparse (just two nonzero entries per equation), we use Eigen's **conjugate gradient solver** for sparse least-square problems [9].

If a tile's position residual error exceeds an absolute and relative error, it is deemed an outlier. If more than one tile position is considered an outlier, the one having the highest residual error is adjusted in each iteration. Adjusting means replacement of a tile's translation by the next highest correlation peak for that tile pair. If all possible translation candidates are exhausted, the tile position is given negligibly small weight ($c_i \rightarrow 0$), thus effectively eliminating it from the global optimization. The iterative process stops once no tile position is considered an outlier.

Both the absolute and relative thresholds are adjustable by the user. The default absolute threshold for the residual error is 1 pixel size, while the default relative threshold is 3 standard deviations, $t_i < 3\sigma_{t_i}$. This relative comparison uses translation differences (right-hand side of the linear system being optimized), not residual errors. This approach caters to typical microscope stages that usually have consistent deviations.

Tile Blending

Smooth blending is achieved by weighting each pixel's contribution by distance from the tile's edge. A synthetic example given in Fig. 4.

If a certain pixel of the output montage is being covered by 3 input tiles, its color is calculated as $c_p = \frac{c_1 d_1 + c_2 d_2 + c_3 d_3}{d_1 + d_2 + d_3}$, where c_i are colors of corresponding pixels of the input tiles and d_i are distances of those pixels from the input tiles' closest edge.

If sub-pixel precision is not requested, or the transform happens to align all the input tiles to integer pixel coordinates (with $\epsilon = 0.0001$ of pixel size), interpolation is not used. Otherwise, linear interpolation is performed, which provides a good performance / accuracy trade-off. The software also provides for B-spline and truncated windowed-sinc interpolation, methods that provide higher intensity fidelity at the cost of computation time.

Results

The `ITKMontage` software module was implemented using high-quality development practices, such as code reviews and abundant testing. We implemented automated unit tests with both real and synthetic data, where the synthetic data consists of artificially constructed tiles having known 'ground truth'. Additionally, we implemented continuous integration testing. Automated tests use tile

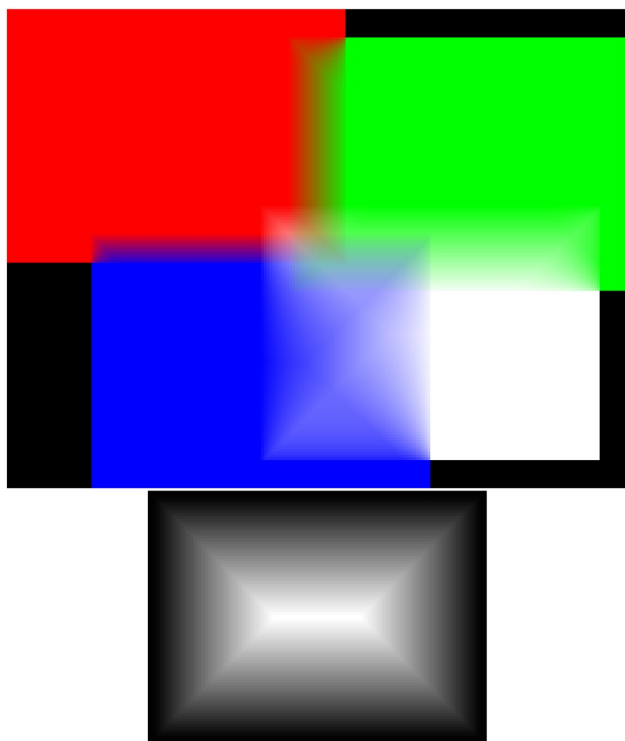


Fig. 4 Blending. Top: smooth blend of 4 uniformly colored, overlapping rectangles. Bottom: overlapping regions of tiles are weighted by the Euclidean distance from the tile's edge

configuration files compatible with [ImageJ/FIJI mounting tools](#) [19].

A caching mechanism can reuse image FFTs for improved performance. Support for image streaming (processing an image piece-by-piece instead of all at once) can reduce peak memory usage when operating on very high resolution images. Stitching is performed in two stages, both capable of streaming. The first stage registers the tiles, with tile translations found by loading a limited number of tiles into memory at a given time. The second stage can generate the output montage image by resampling and blending the resulting montage on a region-by-region basis.

Computational performance is primarily determined by the FFT implementation used for the frequency-domain based phase correlation estimation. Per ITK FFT infrastructure, actual FFT computation can be delegated to either the Visual Numerics Library (VNL) that is bundled with ITK, the Intel Matrix Kernel Library (MKL), the Fastest Fourier Transform in the West (FFTW) library, or CUDA FFT (cuFFT).

ITKMontage is available as a [remote module](#) in [Insight Toolkit 5.1.0](#) and later. The provided GitHub link includes the test data featured in this paper. We have also integrated this montaging software module into an end-user application primarily oriented for material science, [DREAM.3D](#) [8], Fig. 11.

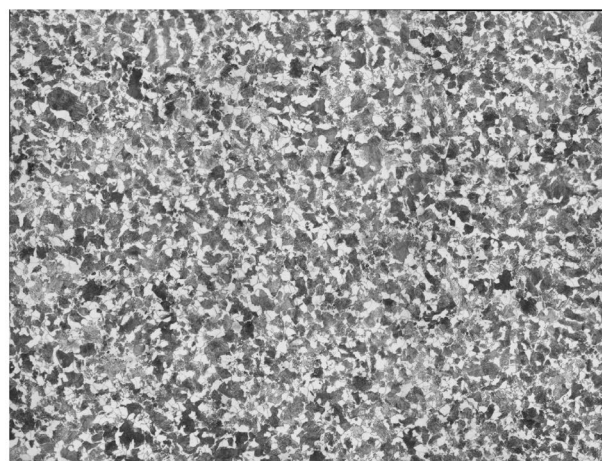
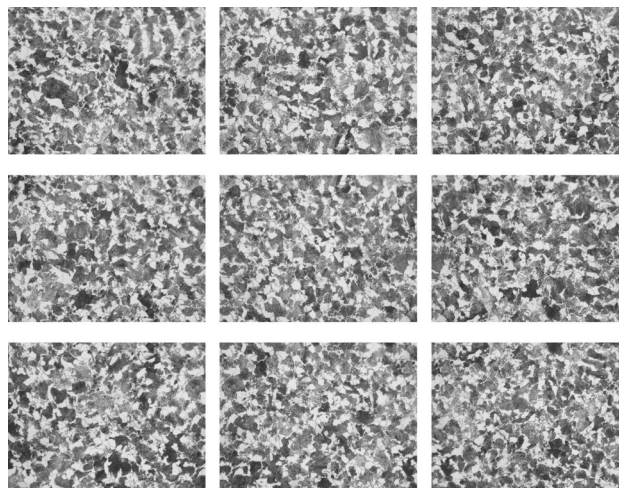


Fig. 5 Montage example: medium carbon steel (data set name: MCS), 3×3 tiles, overlap 11% (top). Final size 3588×2706 pixels (bottom)

Qualitative Results

We evaluated our stitching software on material science microscope images, shown in Figs. 1, 2, 3, 4, 5, 6, 7, and 8. However, the tool works well for other types of data, e.g., Figs. 9 and 10. Visual inspection of these Figures shows high-quality montages without visible stitch lines.

Quantitative Results

As ground truth for an image montage is not generally known, we developed a generator that takes an input image and splits it into a specified number of tiles with overlap. The simulated positions of tiles are allowed to vary, approximating real-world conditions. The chosen tiles are resampled from the original image using windowed-sinc interpolation. The heterogeneity bias was not simulated, partly because of the low impact it has on sub-pixel accuracy, and partly because of the effort required for implementation.

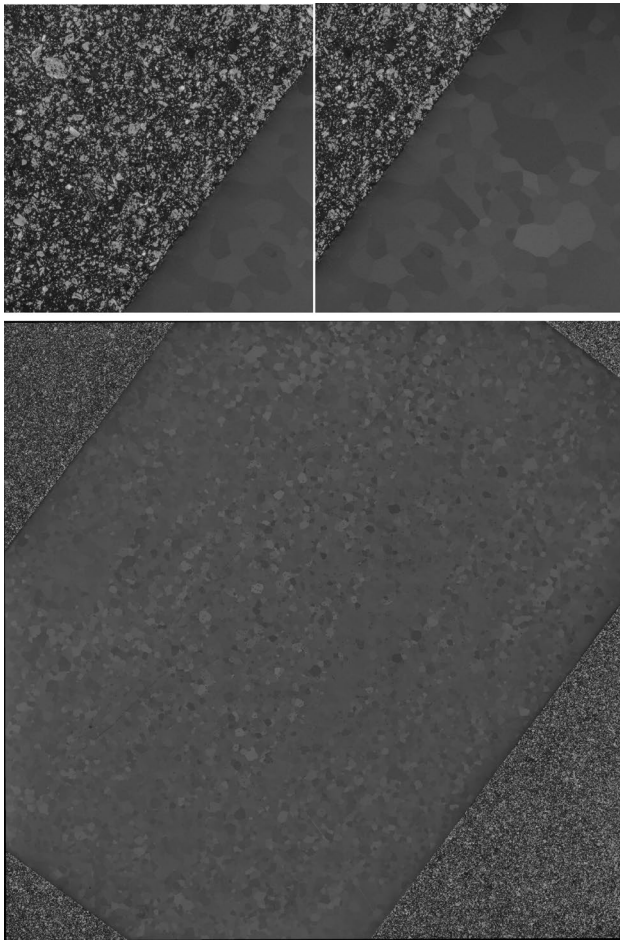


Fig. 6 Montage example: titanium alloy Ti-7Al, 18 × 18 tiles, overlap 50% (top). Final size 9155 × 9139 pixels (bottom)

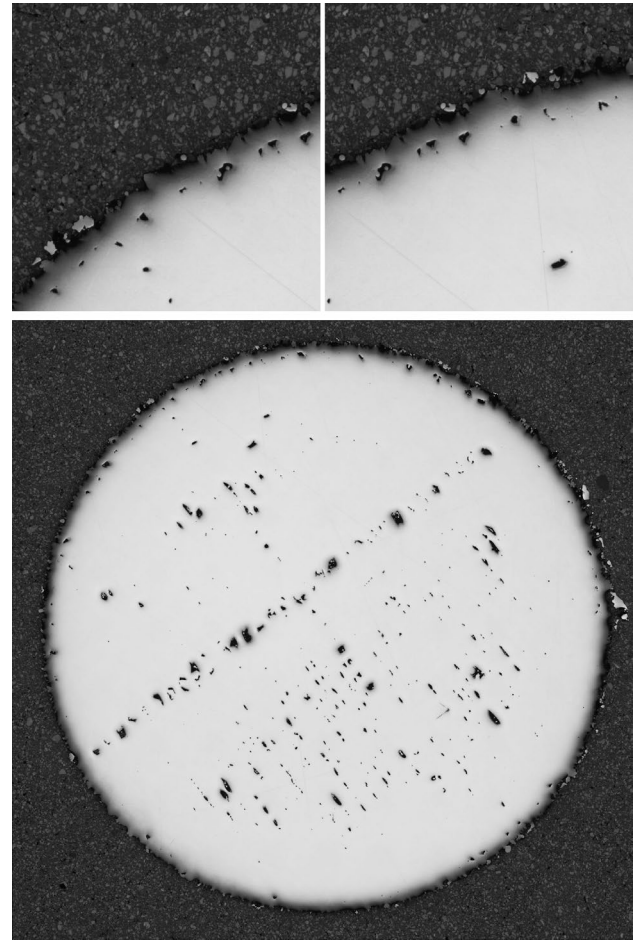


Fig. 7 Montage example: titanium alloy Ti-6Al-4V, 10 × 10 tiles, overlap 50% (top). Final size 5224 × 5221 pixels (bottom)

Testing reveals that [ImageJ/FIJI montaging tools](#) [19] usually has slightly better sub-pixel accuracy (Figs. 13 and 14). But it is an order of magnitude slower on the test computer, probably because the code is single-threaded. Our implementation performs well even when using the single-threaded FFT implementation provided by the Visual Numerics Library (VNL) that comes bundled with ITK. It performs similarly when using the Intel Matrix Kernel Library (MKL), since our code is multi-threaded and test cases consist of a sufficient number of tiles to fully exploit the parallelism of the test computer. The test computer utilized the following configuration: AMD Ryzen 7 PRO 1700 CPU (8 cores, 16 threads) @ 3.0 GHz with 32 GB of RAM.

The ground truth tests and parameters used are provided in Table 1. Table 2 and the charts of Figs. 13 and 14 below show accuracy. Table 4 presents execution speed.

It is interesting to note that in most cases the pairwise registration error (Table 3) is less than total error (pairwise

+ global optimization). This suggests that global optimization step usually leads to compounding of errors, not their annihilation. It is therefore the prime target for possible future improvement.

We used default options for both ITKMontage and ImageJ, see Fig. 12.

Figures 13 and 14 show norm of deviations from ground truth positions of ImageJ/Fiji and ITKMontage, where the X-axis is tile number and Y-axis is its discrepancy with respect to ground truth. Y-axis units are whole pixels. Most deviations are under a pixel. For ITKMontage, four peak interpolation algorithms are shown. Integer values mean no interpolation. Parabolic (the default), cosine, and weighted mean phase interpolation are shown in Fig. 3.

For generating these results, we used Fiji version 1.52p, ITK revision 67eb78ed from 2020-07-10 and ITKMontage revision b1e41dd from 2020-07-24.

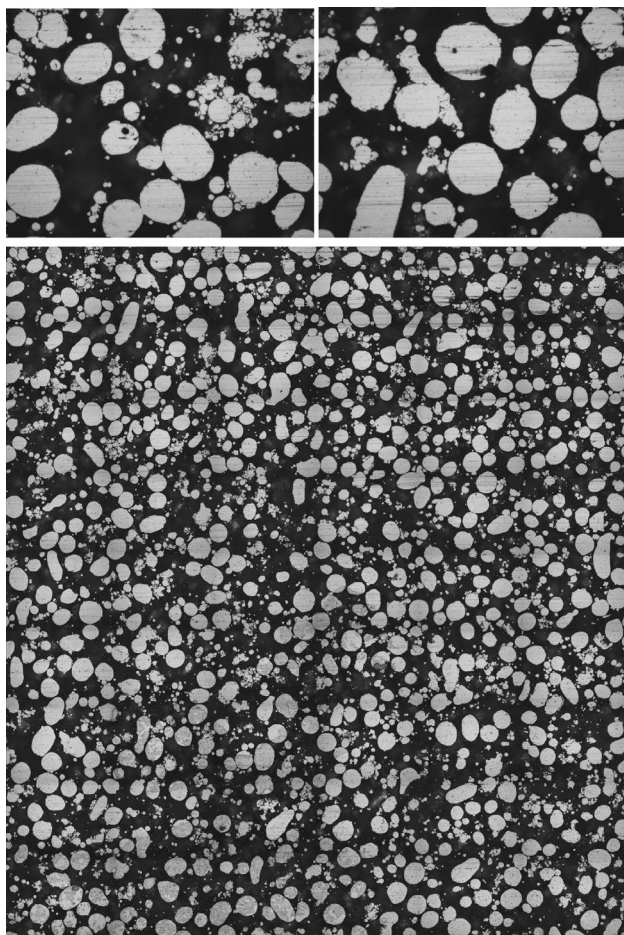


Fig. 8 Montage example: composite (data set name: MNML5), 6×9 tiles, overlap 5% (top). Final size 7442×8334 pixels (bottom). 0.2079 microns/pixel

Selected Limitations

While ITKMontage provides state-of-the-art capabilities for leveling and stitching montage images, the stitching portion of the tool operates by computing relative tile *translations* alone. There are no provisions for correcting the myriad noise or distortions often found for optical, electron-optical, and digital imaging systems. For managing these distortions, we recommend the user correct the individual image tiles prior to montage stitching [2, 15].

Other limitations pertain to image modes and data depth that are common in the physical sciences. Specifically, when one collects electron backscattered diffraction (EBSD) maps, or chemical spectra maps such as from energy dispersion spectroscopy (EDS), the underlying crystallographic or chemical spectra data must be transformed in addition to the grayscale or color image map. ITKMontage has no provisions for stitching such data; however, selected examples of tools that correct these data sets are emerging, and may

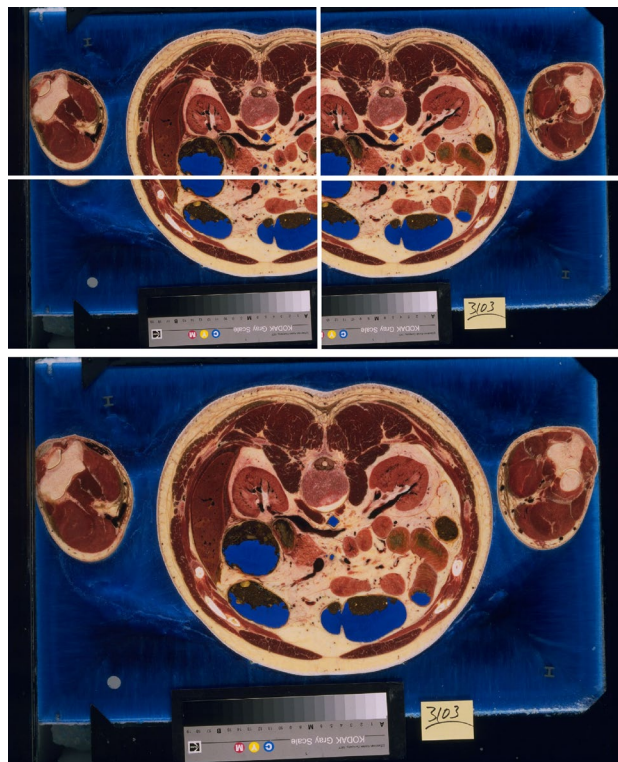


Fig. 9 Montage example: visible human male (data set name: VisibleHumanMale1608), 2×2 tiles, overlap 40% along X, 10% along Y (top). Final size 4096×2700 pixels (bottom)

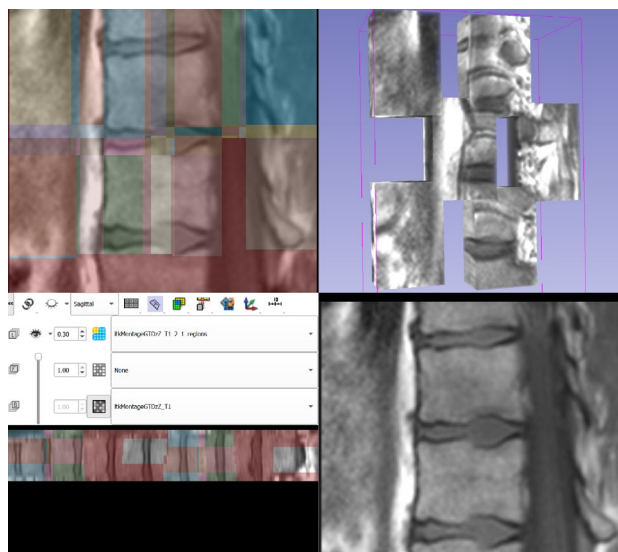


Fig. 10 MRI ground truth example. (data set name: DzZ_T1). Shown here in configuration of $4 \times 8 \times 3$ tiles, overlap 25%, 15% and 50% along the X, Y and Z axes. Original size $144 \times 470 \times 12$ pixels. Tile sizes $44 \times 68 \times 6$ pixels. Top-left: part of XY cross-section overlaid with randomly colored contributing regions. Top-right: 3D rendering of some tiles to show overlap along X and Y axes. Bottom-left: a complete YZ cross-section. Bottom-right: XY cross-section without overlap

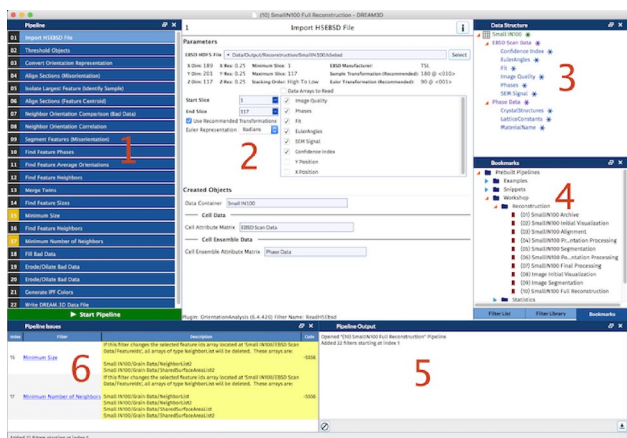


Fig. 11 Screenshot of DREAM.3D end-user application

Table 1 Ground truth test case information

Ground Truth	OMC	MCS	MNML5	VHM	DzZ
Tiles	8 × 6	11 × 3	6 × 9	5 × 4	3 × 6 × 3
PixelCountX	3547	3588	7447	4096	144
PixelCountY	2666	2706	8346	2700	471
PixelCountZ					12
OverlapX	10%	20%	5%	15%	25%
OverlapY	10%	20%	5%	15%	15%
OverlapZ					50%
Note	16-bit	16-bit	Color	Color	16-bit

Table 2 Average pixel error for ground truth test cases

	OMC	MCS	MNML5	VHM	DzZ
Fiji	0.07	0.14	0.14	0.06	3.22
Integer	0.28	0.37	0.32	0.35	0.72
Parabolic	0.12	0.15	0.19	0.03	0.44
Cosine	0.22	0.28	0.21	0.24	0.61
WM phase	0.33	0.40	0.35	0.25	0.73

Bold signifies the best result for the data set

Table 3 Average pairwise registration error

	OMC	MCS	MNML5	VHM	DzZ
Integer	0.29	0.26	0.31	0.31	0.26
Parabolic	0.06	0.05	0.06	0.05	0.14
Cosine	0.21	0.19	0.23	0.22	0.21
WM phase	0.34	0.28	0.35	0.35	0.35

Bold signifies the best result for the data set

Table 4 Running time in seconds for ground truth test cases. All times include reading, registration, optimization, generation of the output montage and writing

	OMC	MCS	MNML5	VHM	DzZ
ITK+MKL	1.5	1.5	2.8	0.7	2.9
ITK+VNL	1.4	1.6	2.7	0.8	2.9
Fiji	11.1	9.5	58.7	11.8	37.7

Bold signifies the best result for the data set

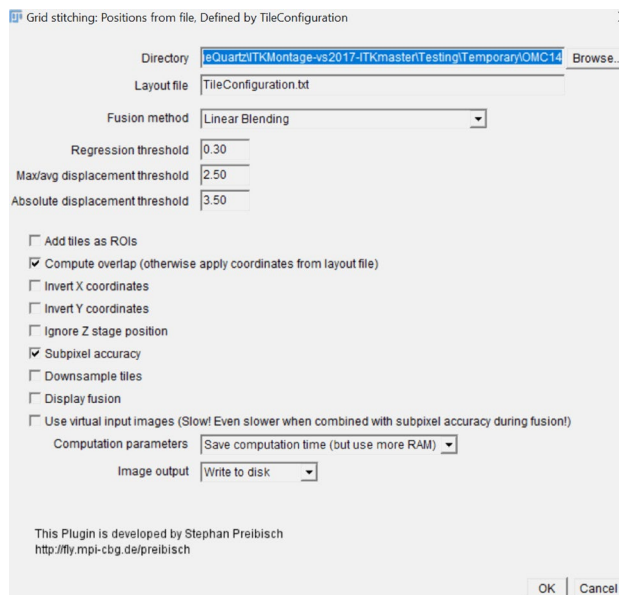


Fig. 12 Options used for stitching plugin in Fiji

one day be incorporated into montage stitching tools [5, 6, 13, 26].

Conclusion

We implemented an open-source, multidimensional scientific image montaging software, and tested it with 2D and 3D inputs. We provide source code examples of how to use it. The main example has support for noise reduction and bias correction. The employed bias correction algorithm relies on maximization of frequency content in the image.

The accuracy of ITKMontage and its DREAM.3D implementation are comparable to that of ImageJ plugin *Image Stitching*, while being about tenfold faster.

The core implementation features coarse-grained parallelism to allow better processor utilization by overlapping different kinds of processing: input-output bound, memory-bound and CPU-bound. Memory usage can be limited

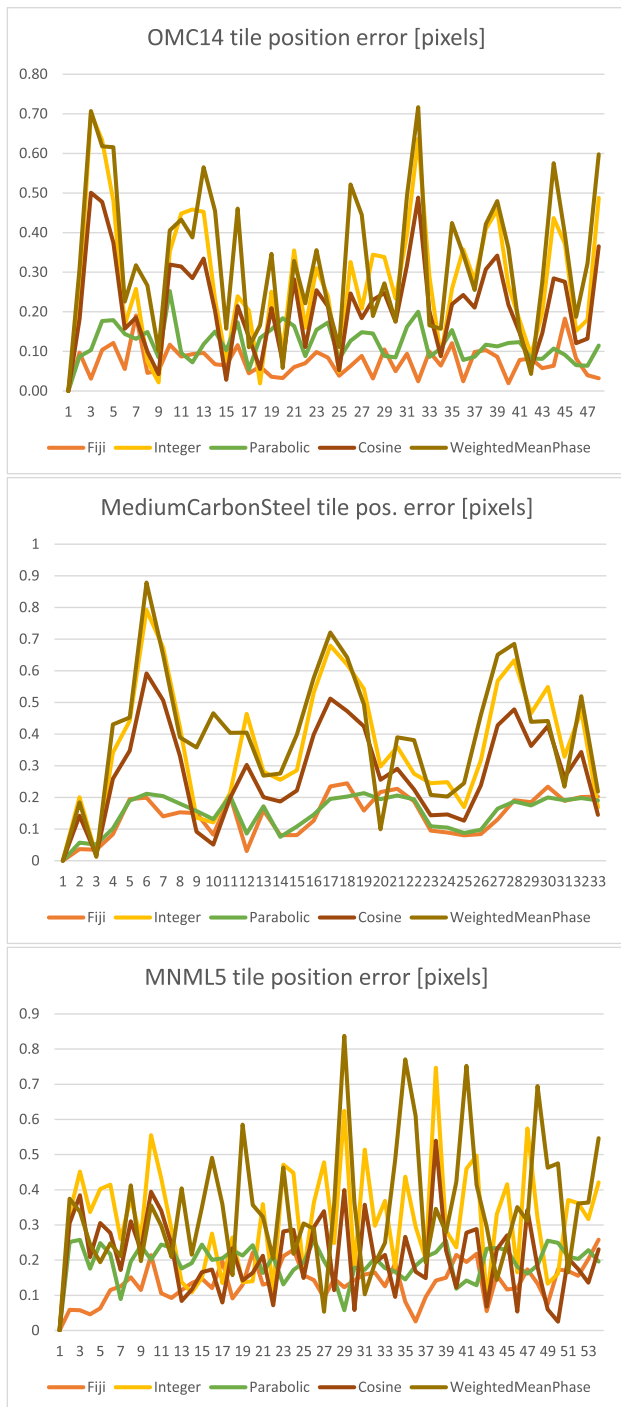


Fig. 13 Accuracy for OMC14 from Fig. 1, Medium Carbon Steel from Fig. 5 and MNML5 from Fig. 8

through loading tiles from disk, on-demand. The tile merging (blending) filter is capable of streaming, which makes it possible to generate extremely large montages without exhausting system memory.

Comprehensive, automated regression tests were added to reproduce expected behavior, according to software

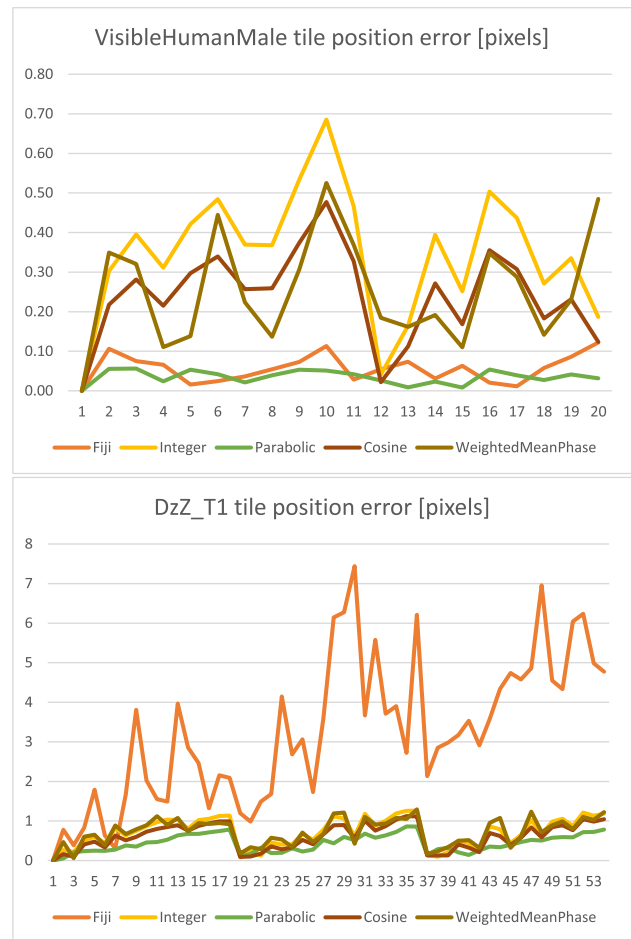


Fig. 14 Accuracy for Visible Human from Fig. 9 and DzZ_T1 from Fig. 10. For the 3D case, the errors accumulate with increasing tile numbers, most noticeably along Y coordinate, least along Z. As tile numbers are linearized, increase in Z with decreases in X and Y creates a noticeable stair shape in error magnitude

quality and open science best practices. The ITKMontage C++ module is wrapped for use from Python, and available in the DREAM.3D graphical application and command line executable interfaces. As a result, large montage stitching is possible from a variety of sources.

Acknowledgements Supported by AFRL contract: *Develop Flexible and Generic Ingest, Archival and Fusion Tools for Air Force Relevant Inspection Data for Holistic Material State Awareness*. Subcontract: FY-2017-TO4-0001. BlueQuartz Prime Contract: FA8650-15-D-5231, Task Order 04. The authors would like to acknowledge Michael Uchic (AFRL) for collecting the OMC data set; Michael Chapman (UES, Inc.) and Michael Uchic for collecting the Ti-7Al and Ti-6Al-4V data sets; and Jonathan Spowart for collecting the MNML5 data set.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflicts of interest associated with this manuscript.

References

- Bican J (2006) Phase correlation method for itk. <https://www.insight-journal.org/browse/publication/138>
- Capitani GC, Oleynikov P, Hovmöller S, Mellini M (2006) A practical method to detect and correct for lens distortion in the tem. *Ultramicroscopy* 106(2):66–74
- Chalfoun J, Majurski M, Bhadriraju K, Lund S, Bajcsy P, Brady M (2015) Background intensity correction for terabyte-sized time-lapse images. *J Microsc* 257(3):226–237
- Chalfoun J, Majurski M, Blattner T, Bhadriraju K, Keyrouz W, Bajcsy P, Brady M (2017) MIST: accurate and scalable microscopy image stitching tool with stage modeling and error minimization. *Sci Rep* 7(1):4988
- Charpagne MA, Strub F, Pollock TM (2019) Accurate reconstruction of ebsd datasets by a multimodal data approach using an evolutionary algorithm. *Mater Charact* 150:184–198
- Chen Z, Lenthe W, Stinville J, Echlin M, Pollock T, Daly S (2018) High-resolution deformation mapping across large fields of view using scanning electron microscopy and digital image correlation. *Exp Mech* 58(9):1407–1421
- Echlin MP, Straw M, Randolph S, Filevich J, Pollock TM (2015) The tribeam system: femtosecond laser ablation in situ sem. *Mater Charact* 100:1–12
- Groeber M.A, Jackson M.A (2014) DREAM.3D: a digital representation environment for the analysis of microstructure in 3D. *Integr Mater Manuf Innov* 3(1):56–72. <https://doi.org/10.1186/2193-9772-3-5>
- Guennebaud G, Jacob B, et al (2010) Eigen v3. <http://eigen.tuxfamily.org>
- Gurumurthy A, Gokhale AM, Godha A, Gonzales M (2013) Montage serial sectioning: some finer aspects of practice. *Metallogr Microstruct Anal* 2(6):364–371
- Ibanez L, Lorensen B, McCormick M, King B, Blezek D, Johnson H, Lowekamp B, Jomier J, Miller J, Lehmann G, Cates J, Ng L, Kim J, Gelas A, Malaterre M, Krishnan K, Hoffman B, Williams K, Budin F, Legarreta JH, Schroeder WR, Aylward S, Zuki D, Liu X, Avants B, Dekker N, Noe A, Popoff M, Hart G, McBride S, Sundaram T, Gouaillard A, Stauffer M, Enquobahrie A, Tustison N, Pathak S, Cedilnik A, Chen T, Shelton D, Helba B, Quammen C, Jin Y, Padfield D, Vercauteren T, Jae Kang H, Turek M, Tamburo R, Audette M, Hernandez-Cerdan P, Foskey M, Hughett P, Doria D, Kindlmann G, Cole D, Fillion-Robin, JC, Turner W, Chen SS, Fonov V, Tasdizen T, Duda J, Galeotti J, Barre S, Aume S, Mosaliganti K, Chandra P, Ghayoor A, Mackelfresh A, Mullins C, Zhuge Y, Martin K, Xue X, Straing M, Estepar R, Squillacote A, Wyman B, Chang W, Guyon JP, Botha C, Baghdadi L, Maekclena Vigneault DP, Awate S, Reynolds P, Rit S, Pincus Z, Finet J, Venkatram R, Zygmunt K, Rondot P, Davis B, Antiga L, Coursolle M, Roden M, Park S, Cheung H, Aaron Cois C, Gandel L, Kaucic R, Yaniv ZD, Hanwell M, Magnotta V, Bertel F, Greer H, Hipwell, J, Chandrashekhara RC, Bigler D, Straing M, Styner M, Robbins S, Chalana V, Le Poul Y, Neundorf A, Lamb P, Gerber S, Williamson Z, Braun-Jones T, Wasem A, Rannou N, Beare R, Members IC (2020) Insightsoftwareconsortium/itk: Itk 5.1.0. <https://doi.org/10.5281/zenodo.3888793>
- Kuglin CD, Hines DC (1975) The phase correlation image alignment method. In: *Proceedings of the International Conference Cybernetics Society*, pp. 163–165
- Levinson AJ, Rowenhorst DJ, Lewis AC (2015) Automated image alignment and distortion removal for 3-D serial sectioning with electron backscatter diffraction. *Microsc Microanal* 21(S3):1673–1674
- Likar B, Maintz J, Ma Viergever, Pernuš F (2000) Retrospective shading correction based on entropy minimization. *J Microsc* 197(3):285–295
- Maraghechi S, Hoefnagels J, Peerlings R, Rokoš O, Geers M (2019) Correction of scanning electron microscope imaging artifacts in a novel digital image correlation framework. *Exp Mech* 59(4):489–516
- Mazzamuto G, Silvestri L, Costantini I, Orsini F, Roffilli M, Frascioni P, Sacconi L, Pavone F (2018) Software tools for efficient processing of high-resolution 3d images of macroscopic brain samples. In: *Biophotonics Congress: Biomedical Optics Congress 2018 (Microscopy/Translational/Brain/OTS)*, p. JTh3A.64. *Optical Society of America* (2018). <https://doi.org/10.1364/TRANSLATIONAL.2018.JTh3A.64>. <http://www.osapublishing.org/abstract.cfm?URI=BRAIN-2018-JTh3A.64>
- McCormick M, Liu X, Ibanez L, Jomier J, Marion C (2014) ITK: enabling reproducible research and open science. *Front Neuroinf* 8:13
- McCormick MM, Varghese T (2013) An approach to unbiased subsample interpolation for motion tracking. *Ultrason Imaging* 35(2):76–87
- Preibisch S, Saalfeld S, Tomancak P (2009) Globally optimal stitching of tiled 3D microscopic image acquisitions. *Bioinformatics* 25(11):1463–1465
- Rowenhorst DJ, Nguyen L, Murphy-Leonard AD, Fonda RW (2020) Characterization of microstructure in additively manufactured 316l using automated serial sectioning. *Curr Opin Solid State Mater Sci* 100819
- Sled JG, Zijdenbos AP, Evans AC (1998) A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Trans Med Imaging* 17(1):87–97
- Svilainis L (2019) Review on time delay estimate subsample interpolation in frequency domain. *IEEE Trans Ultrason Ferroelect Freq Control* 66(11):1691–1698
- Tustison N, Gee J (2010) N4itk: Nick's N3 ITK implementation for MRI bias field correction
- Tustison NJ, Avants BB, Cook PA, Zheng Y, Egan A, Yushkevich PA, Gee JC (2010) N4ITK: improved N3 bias correction. *IEEE Trans Med Imaging* 29(6):1310–1320
- Uchic MD, Groeber MA, Rollett AD (2011) Automated serial sectioning methods for rapid collection of 3-D microstructure data. *JOM* 63(3):25–29
- Wang Y, Suyolcu YE, Salzberger U, Hahn K, Srot V, Sigle W, van Aken PA (2018) Correcting the linear and nonlinear distortions for atomically resolved stem spectrum and diffraction imaging. *Microscopy* 67(1):i114–i122